Knowledgeable In-Context Tuning: Exploring and Exploiting Factual Knowledge for In-Context Learning

Anonymous ACL submission

Abstract

Large pre-trained language models (PLMs) enable in-context learning (ICL) by conditioning on a few labeled training examples as a text-based prompt, eliminating the need for parameter updates and achieving competitive performance. In this paper, we demonstrate that *factual knowledge* is imperative for the performance of ICL in three core facets, i.e., the inherent knowledge learned in PLMs, the factual knowledge derived from the selected in-context examples, and the knowledge biases in PLMs for output generation. To unleash the power of large PLMs in few-shot scenarios, we introduce a novel **K**nowledgeable **I**n-**C**ontext **T**uning (KICT) framework to further improve the ICL's performance: 1) injecting knowledge to PLMs during continual self-supervised pre-training, 2) judiciously selecting the examples with high knowledge relevance, and 3) calibrating the prediction results based on prior knowledge. We evaluate the proposed approaches on auto-regressive models (e.g., GPT-style PLMs) over multiple text classification and question answering tasks. Experiments results demonstrate that KICT substantially outperforms strong baselines, and improves by more than 13% and 7% on text classification and question answering tasks, respectively. [1]

1 Introduction

Pre-trained language models (PLMs) have become the imperative infrastructure in the natural language processing (NLP) community (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019). To enable large PLMs to perform well without any parameter updates, in-context learning (ICL) has become one of the flourishing research topics in many few-shot NLP tasks, which aims at generating the prediction of the target example by conditioning on a few labeled samples (Brown et al., 2020). As shown in Figure 1, the key component of ICL is the text-based prompt that serves as the demonstration.

[1]All codes and datasets will be released upon acceptance.

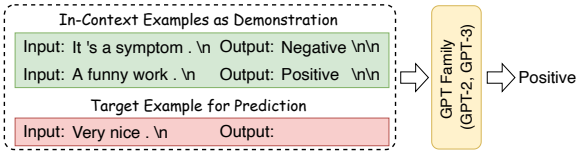


Figure 1: An example of in-context learning (ICL).

Previous works have explored multiple aspects that affect the performance of ICL (Dong et al., 2023), such as input-output mapping (Min et al., 2022b; Kim et al., 2022), extensive data resources (Mishra et al., 2022; Chen et al., 2022b; Min et al., 2022a), and prediction calibration (Zhao et al., 2021). Liu et al. (2022); Lu et al. (2022) have explored some others, such as the prompt format (e.g., "Input:", "Output:"), the selection of labeled data, and example permutation. However, these works ignore the influence of *factual knowledge* in ICL, which is one of the non-negligible factors in the era of NLP (Hu et al., 2022).

To this end, we explore the effectiveness of ICL from the perspective of *factual knowledge*. As seen in Figure 2, when randomly replacing or removing entities and labels from text-based prompts, the average accuracy decreases significantly. The destruction performance is also universal across model scales. In further analysis, we discover that: 1) more intrinsic factual knowledge learned in the pre-training stage is typically beneficial to the PLMs to improve its effectiveness. 2) The factual knowledge (e.g., entities and labels) derived from selected in-context examples is key to the performance of ICL. 3) The PLMs tend to generate common words which may have high frequencies in the training corpora, resulting in biased prediction.

After analyzing these knowledge facets, a natural question arises: *how to fully employ factual knowledge to further improve the performance of ICL?* To reach this goal, we focus on casual auto-regressive PLMs (e.g., GPT-2 (Radford et al., 2019) and OPT (Zhang et al., 2022a)) and present a novel

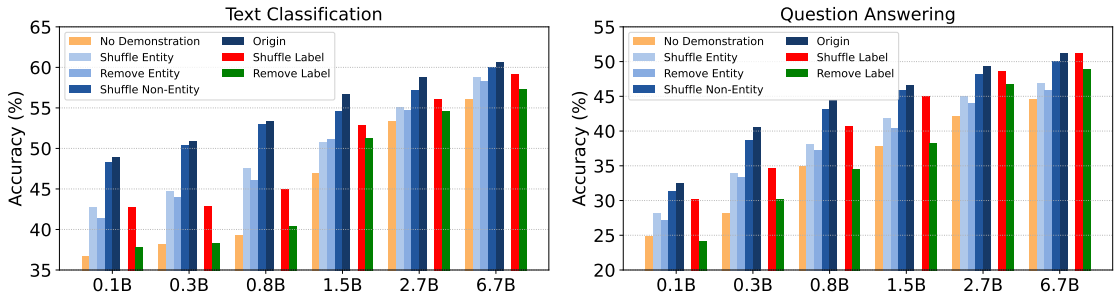


Figure 2: Results of different scales of GPT-2 and OPT models over 8 text classification tasks and 4 question answering tasks when using different component destruction settings. Each target example has $K = 8$ labeled samples as the demonstration. Results indicate that factual knowledge is crucial to ICL's performance.

Knowledgeable **I**n-**C**ontext **T**uning (KICT) framework, which involves the knowledgeable guidance in *pre-training*, *prompting*, and *prediction* of these models. Specifically, to endow the PLMs with text generation abilities by better probing inherent knowledge, we introduce several knowledgeable self-supervised tasks to inject knowledge into PLMs during the *pre-training* stage. For text-based *prompting*, we propose a knowledgeable example retrieval algorithm to judiciously select in-context examples which have relevant knowledge with the target example. Finally, during *prediction*, we utilize the knowledge-wise prior of label words from a underlying Knowledge Base (KB) to calibrate the prediction distributions derived from PLMs. Each of the proposed techniques is plug-and-play and can be freely combined together, facilitating the users to exploit knowledge for improving ICL.

To evaluate the effectiveness of the KICT framework, we employ auto-regressive PLMs (e.g., GPT-style models) to conduct extensive experiments over multiple classification and question answering tasks. Results demonstrate that each proposed procedure achieves substantial improvements.

2 The Impact of Knowledge on In-Context Learning

In this section, we aim at investigating whether *factual knowledge* affects the performance of ICL.

Preliminary experimental settings. We follow Min et al. (2022b) and Kim et al. (2022) to perform empirical experiments by component destruction. Specifically, given one target example text X^{tgt}, we randomly select K training samples $\widetilde{\mathcal{D}} = \{(X_i^{trn}, y_i^{trn})\}_{i=1}^{K}$ to form a text-based prompt. We identify all entities in the prompt, and then design some destruction settings as follows. 1) `Shuffle Entity` means to randomly replace all entities with others in the KB. 2) `Shuffle Non-Entity` denotes replacing some non-entity words (e.g., "It", "have") with others in the PLM vocabulary. 3) `Shuffle Label` represents replacing all the golden labels with wrong labels. 4) `Remove Entity` and `Remove Label` aim to remove all entities and labels from the prompt, respectively. 5) `No Demonstration` is a typical zero-shot method that does not use any labeled data (Min et al., 2022b). We choose different scales of GPT-2 (0.1B-1.5B) and OPT (Zhang et al., 2022a) (2.7B-6.7B) to evaluate 8 text classification tasks and 4 question answering tasks. [2] In default, we randomly sample $K = 8$ labeled samples for each task and run experiments with 5 different random seeds. More details can be found in Appendix A. The findings are summarized below.

The inherent knowledge in the PLM itself is beneficial to the performance of downstream tasks. As in Figure 2, models can obtain remarkable few-shot performance when increasing the scale size. We hypothesize that this is because larger models can learn more valuable semantics in the pre-training corpus. To validate this assumption, we do not use any text-based prompts to perform zero-shot inference (i.e., `No Demonstration`). In other words, only the intrinsic knowledge learned during pre-training can provide the model guidance on the prediction. We can see that the performance gap between 6.7B and 0.1B is about 20% on both text classification and question answering tasks. This suggests that the inherent knowledge learned during pre-training is imperative (Yang et al., 2021).

The factual knowledge in selected in-context examples is key to ICL. As shown in Figure 2, the original setting (`Origin`) outperforms others

[2] We do not use larger GPT-style models (e.g., GPT-3 (Brown et al., 2020)) due to resource constraints. Yet, our findings are generally consistent across different model scales.

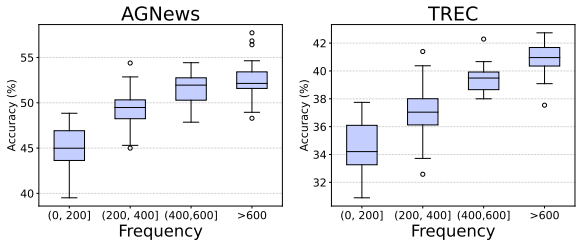


Figure 3: 4-shot results of GPT-2 (urge) over AGNews and TREC. For each frequency region, we sample top-5 label words for each category and report the accuracy for all label mapping permutations.

in each model scale. We find that changing the non-entities does not significantly reduce the performance, while replacing or removing the entities will decrease the average accuracy a lot in both text classification and question answering tasks. This indicates that factual knowledge in text-based prompts is the key factor for the PLM to understand the task. Further, we also find the label is also imperative for ICL, where similar findings are presented in (Kim et al., 2022). Different from Min et al. (2022b), we suggest that labels can also be viewed as one of the knowledge that guides the PLM to perceive semantics during model inference.

PLMs tend to generate common label words due to knowledge bias. To test whether the prediction suffers from bias problems, we choose two knowledge-intensive tasks (i.e., AGNews (Zhang et al., 2015), and TREC (Voorhees and Tice, 2000)). We first obtain top-5 predictions at the output position for each training example [3] and calculate the frequency statistics of each generated label word. We then choose 4 labeled examples from the training set. For each category, we randomly select 2 label words from each frequency region and report the average accuracy of all label mapping permutations [4]. Results in Figure 3 show that the performance highly relies on the label word frequency, which indicates that the frequency of factual knowledge learned in PLMs is crucial to the prediction [5].

3 The Proposed KICT Framework

The preliminary experiments demonstrate that *factual knowledge* has a substantial effect on ICL. This suggests that we can fully exploit the knowledge to boost the performance in various of processes in ICL, including *pre-training*, *prompting*, and *prediction*. To reach this goal, we introduce KICT, a novel **K**nowledgeable **I**n-**C**ontext **T**uning framework to better exploit knowledge to unleash the PLM power towards answer generation. In this framework, we introduce knowledgeable pre-training (KPT) with three well-designed self-supervised tasks to inject factual knowledge into the PLMs. Then, we present a knowledgeable example retrieval (KER) algorithm to judiciously select knowledge-relevant in-context examples. At last, a knowledgeable prediction calibration technique (KPC) is used to calibrate the prediction distribution via prior information derived from KB. The framework overview is shown in Figure 4.

3.1 Knowledgeable Pre-Training

This part describes three knowledge-aware self-supervised learning tasks to inject factual knowledge into the PLM, i.e., *Masked Entity Prediction*, *Entity Description Generation*, and *Knowledge Question Answering*. Different from Chen et al. (2022a), we aim at leveraging an external KB to enrich the language generation abilities w.r.t. important entities. Hence, the input is a training corpus $\{X\}$ and a KB $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is a set of entities, $\mathcal{R}$ is a set of relations, and $\mathcal{T}$ is a set of triples expressing factual knowledge.

Masked Entity Prediction (MEP). This task requires the model to predict the missing entities in the text to learn explicit knowledge, which is similar to the *Masked Language Modeling* in BERT-style PLMs (Devlin et al., 2019; Liu et al., 2019). Concretely, given one piece of text tokens $X = \{x_i\}$, we recognize all entities $E_X = \{e | e \in \mathcal{G}, e \in X\}$ via entity linking toolkit, where $e = \{x_j | x_j \in X\}$ is an entity with multiple tokens. For each entity e, 50% of the time is replaced with special tokens (e.g., "_"), while another 50% of the time is replaced with random tokens . Thus, we can obtain a training example $\hat{X} = \{\hat{x}_i\}$. We generate a label mask vector $\mathcal{M}_{\hat{X}}$ to represent what position is used for training[6], and $\mathcal{M}_{\hat{X}_i} = \mathbb{I}(\hat{x}_i \in E_X)$, where $\mathbb{I}(\cdot)$ is the indicator function.

Entity Description Generation (EDG). This task aims to generate a text description step by step based on the given entities. Specifically, given one text X and a corresponding entities set E_X, we

[3] The scale of the training set is larger than the testing set so that the statistics can be more obvious.

[4] Take AGNews as an example, it has 4 classes and each has 2 label words, there are $2^4 = 16$ label mapping permutations.

[5] Similar findings are also described in (Zhao et al., 2021).

[6] The word in red in Figure 4 (left).

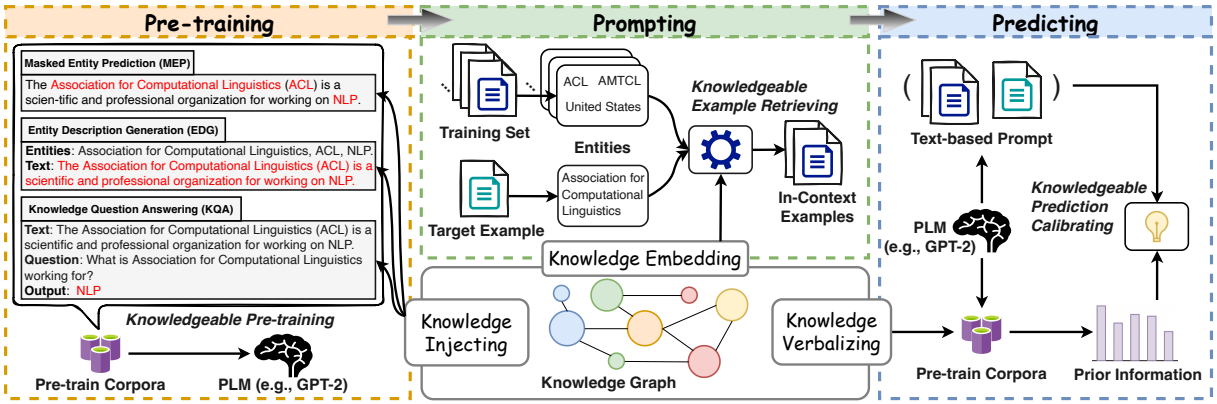


Figure 4: The overview of KICT framework. We introduce multiple plug-and-play knowledgeable techniques for better exploiting knowledge to improve ICL's performance. **Left**: We present three knowledge-aware self-supervised learning tasks to inject factual knowledge into PLM during pre-training. **Medium**: We incorporate the knowledge entities to select in-context examples which have high knowledge relevance with the target example. **Right**: For prediction, we obtain the prior information derived from large-scale corpora for calibrating the prediction.

construct a prefix text which is a linearized string formed by the template "Entities:", all entities in E_X and the template "Text:". The suffix text is the original text X. Likewise, we can generate a training example $\hat{X}$ and a label mask vector $\mathcal{M}_{\hat{X}}$ and $\mathcal{M}_{\hat{X}_i} = 1$ when $\hat{x}_i$ is in the suffix string.

Knowledge Question Answering (KQA). To fully use off-the-shelf triples in KB, we also consider a knowledge-aware question answering task which aims to generate the entity based on a question. Specifically, given one text X and a corresponding entities set E_X, we can obtain two entities $e_h, e_t \in E_X$ which have 1-hop relation $r \in \mathcal{R}$ and form a triple $(e_h, r, e_t) \in \mathcal{T}$, where e_h and e_t are the head entity and tail entity, respectively. Inspired by Wang et al. (2022), we design a template for each triple and convert it to a question to ask the model to predict the tail entity, and obtain the training example $\hat{X}$ and the label mask vector. We denote $\mathcal{M}_{\hat{X}_i} = 1$ when $\hat{x}_i$ is the token of the selected tail entity.

During the pre-training process, we randomly select multiple examples from the same task to form a training instance $\mathcal{X} = \{\hat{X}\}$ until reaching the maximum sequence length (i.e., 2048). We calculate the cross-entropy loss at the output position (where $\mathcal{M}_{\hat{X}} = 1$). Formally, we have:

$$\mathcal{L} = \frac{1}{|\mathcal{X}|} \sum_{\hat{X} \in \mathcal{X}} \frac{1}{T_{\hat{X}}} \sum_{\hat{x}_i \in \hat{X}} \mathcal{M}_{\hat{X}_i} \log p(y_i | \hat{X}_{<i}), \quad (1)$$

where y_i is the ground truth. $p(\cdot)$ denotes the prediction probability. $T_{\hat{X}} = \sum_{\hat{x}_i \in \hat{X}} \mathcal{M}_{\hat{X}_i}$ is the number of positions that model needs to calculate the loss.

3.2 Knowledgeable Example Retrieval

Although we have obtained a powerful and knowledgeable PLM, the performance of ICL highly depends on the selection and order of labeled examples (Brown et al., 2020). Previous works (Liu et al., 2022; Lu et al., 2022; Rubin et al., 2022) have investigated that the PLM itself can generate suitable text-based prompts. However, they pay little attention to the tangible value of *factual knowledge* in KB.

We introduce a novel knowledgeable example retrieval (KER) algorithm to incorporate knowledge to select in-context examples. The process is visualized in Figure 4 (medium), and the algorithm is shown in Algorithm 1 in Appendix C. Specifically, given a training set $D_{trn} = \{(X_i^{trn}, y_i^{trn}, E_i^{trn})\}$ and a target set $D_{tgt} = \{(X_j^{tgt}, E_j^{tgt})\}$ (i.e., testing set), where X_i^{trn} and X_j^{tgt} denote the input texts, y_i^{trn} denotes the label of the training example, and E_i^{trn} and E_j^{tgt} are the corresponding entities set. Recall that the knowledge in the text-based prompts is key to ICL. The task of KER aims to choose a set of training examples that have high knowledge relevance to the target set. Hence, a simple way is to retrieve the examples in which the entities can cover more target examples. We utilize Jaccard similarity to calculate the similarity between two examples by $d_{jac}(i,j) = \frac{|E_i^{trn} \cap E_j^{tgt}|}{|E_i^{trn} \cup E_j^{tgt}|}$. Yet, the Jaccard similarities of most example pairs

are zeros, so we further leverage the pre-trained knowledge embeddings to retrieve the training examples which are more *similar* to the target set in the semantic space. Formally, we obtain the averaged representations $\mathbf{e}_i$ and $\mathbf{e}_j$ of all entities in E_i^{trn} and E_j^{tgt}, respectively. The Euclidean distance $d_{sem}(i,j)$ between $\mathbf{e}_i$ and $\mathbf{e}_j$ can be used to represent the difference in the semantic space. Thus, the final knowledge relevance between two examples can be calculated as:

$$d(X_i^{trn}, X_j^{tgt}) = \alpha \frac{d_{jac}(i,j)+\gamma}{\max_{X_k^{trn}\in\mathcal{D}_{trn}} d_{jac}(i,k)+\gamma} + (1-\alpha)(1-\frac{d_{sem}(i,j)}{\max_{X_k^{trn}\in\mathcal{D}_{trn}} d_{sem}(i,k)}), \quad (2)$$

where $0 \leq \alpha \leq 1$ and $\gamma > 0$ are hyper-parameters. For each X_i^{trn}, the sampling weight is:

$$s'(X_i^{trn}) = \frac{s(X_i^{trn})}{\sum_{X_j^{trn}\in\mathcal{D}_{trn}} s(X_j^{trn})}, \quad (3)$$

where $s(X_i^{trn})$ can be computed as:

$$s(X_i^{trn}) = \frac{1}{|\mathcal{D}_{tgt}|} \sum_{X_j^{tgt}\in\mathcal{D}_{tgt}} d(X_i^{trn}, X_j^{tgt}). \quad (4)$$

Intuitively, the training example with high weight means that it has high knowledge relevance with all target examples. Ultimately, we can sample K training examples based on these weights.

3.3 Knowledgeable Prediction Calibration

After model pre-training and in-context example selection, we can directly generate the output for the target example $X^{tgt} \in \mathcal{D}_{tgt}$ by:

$$\hat{y} = \arg\max_{v\in\mathcal{V}} p(y=v|\widetilde{\mathcal{D}}, X^{tgt}), \quad (5)$$

where $\mathcal{V}$ is the verbalizer to map the label words to the corresponding class [7]. $\widetilde{\mathcal{D}}$ is the set of in-context examples. However, we find that the frequency of label words (in the classification task) or entities (in the question answering task) may induce bias in the prediction probability (recall to Section 2). To remedy this dilemma, we aim to leverage the prior information of the label words to calibrate the prediction of each target example. Specifically, we obtain a subset of training corpora $\mathcal{S}$ from the KQA task and calculate the contextualized prior of each candidate label word or entity $v \in \mathcal{V}$ at the output position by:

$$P(v) \approx \frac{1}{|\mathcal{S}|} \sum_{\hat{X}\in\mathcal{S}} p(y=v|\hat{X}), \quad (6)$$

where $\hat{X}$ is the training example, and $P(v)$ denotes the approximated prior information of candidate v. We remove the label word or entity v whose prior probability is smaller than a threshold (Hu et al., 2022). Thus, the output can be upgraded by the calibrated prediction:

$$\hat{y} = \arg\max_{v\in\mathcal{V}} p(y=v|\widetilde{\mathcal{D}}, X^{tgt})/P(v). \quad (7)$$

Remarks. Most recent works (Hu et al., 2022; Zhao et al., 2021) focus on prediction calibration. Different from them, we fully exploit prior knowledge from the large-scale corpus to debias, instead of only utilizing in-domain data or designing task-agnostic content-free input (e.g., "N/A").

4 Experiments

4.1 Implementation Settings and Baselines

For the pre-training corpus, we use Wikipedia Dumps (2020/03/01)[8], which consists of 25,933,196 sentences. Further, the KB we used is WikiData5M (Wang et al., 2021b), which includes $3,085,345$ entities and 822 relation types. By default, we choose GPT-2 (large) with 0.8B parameters as the backbone . For downstream tasks, we consider 8 text classification tasks and 4 question answering tasks. The details of corpora and downstream benchmarks are shown in Appendix B. The implementation details of pre-training, prompting, and prediction can be found in Appendix C.

We consider the following baselines: 1) **In-Context Learning (ICL)** is the vanilla version proposed by GPT-3. 2) **Calibrate Before Use (CBU)** (Zhao et al., 2021) is a typical method that aims to de-bias the prediction via content-free prompts. 3) **KATE** (Liu et al., 2022) uses the CLS embeddings of a RoBERTa-large model as sentence representations, and retrieves the nearest K neighbors for each target example as the final in-context examples. 4) **MetaICL** (Min et al., 2022a) improves ICL by meta-learning the objective of ICL in cross-task settings. 5) **SelfSup.** (Chen et al., 2022a) improves ICL by multiple self-supervised

[7] For classification, $\mathcal{V}$ denotes the label words set. For question answering, $\mathcal{V}$ denotes the whole vocabulary set.

[8] https://dumps.wikimedia.org/enwiki/

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	76.18±7.2	54.46±2.3	56.85±2.4	52.93±3.2	53.94±5.0	42.50±1.8	51.56±4.1	45.67±6.6	54.26
CBU (Zhao et al., 2021)	82.71±4.4	63.07±3.9	57.93±2.8	53.19±3.9	54.87±2.8	51.34±1.7	54.61±3.7	55.42±2.8	59.14
KATE (Liu et al., 2022)	81.33±3.8	58.04±3.9	59.40±2.4	53.57±3.5	53.17±2.7	45.48±2.1	54.69±2.8	50.28±3.4	57.00
MetaICL† (Min et al., 2022a)	87.40±5.0	62.91±2.0	60.22±3.4	55.18±1.9	57.06±2.8	49.20±2.5	56.09±1.8	55.80±2.4	60.48
SelfSup.† (Chen et al., 2022a)	87.94±3.0	62.33±2.0	62.00±2.2	54.77±1.8	57.27±2.6	45.80±2.5	55.59±2.5	57.44±3.2	60.39
KICT†	**91.21±2.9**	**69.96±0.7**	**69.59±1.0**	**60.66±1.2**	**63.74±4.2**	**56.07±3.8**	**63.52±5.5**	**68.89±5.7**	**67.96**
only w. KPT†	90.04±3.5	66.65±1.9	67.39±2.6	58.97±3.0	58.26±3.3	55.43±2.0	60.16±2.2	59.74±4.4	64.58
only w. KER	84.05±2.7	59.26±2.5	59.93±1.0	57.23±1.2	53.79±4.0	51.36±3.8	55.52±5.1	52.70±3.3	59.23
only w. KPC	85.52±3.9	64.77±0.7	63.13±1.2	57.69±2.4	55.94±1.2	54.07±2.8	56.92±2.7	57.24±5.5	61.91

Table 1: The 8-shot performance (%) on GPT-2 (large) of different learning settings with standard deviations over text classification benchmarks. Compared with other baselines, our framework achieves consistent improvement. † denotes the method involves parameters update for ICL. "only w." means we only use one technique in KICT.

learning tasks. We also choose RoBERTa-large to perform fully **Fine-tuning** to demonstrate the ceiling performance of each task.

4.2 Main Results

Table 1 and Table 2 respectively report the results over text classification and question answering tasks in the 8-shot setting. We thus make the following observations: 1) Our proposed framework KICT outperforms strong baselines and achieves substantial improvements over all benchmarks. Specifically, compared with ICL, the averaged result over text classification task is improved by 13.70% , which is larger than other baselines. The average gain over question answering tasks is also more than 7%, despite that there is still room for improvement on unseen target domains, likely because they require more challenging generalization and commonsense ability. 2) Compared with ICL, KER and KCP make great contributions to the performance. Particularly, KER and KCP also respectively outperform strong baselines KATE and CBU, indicating the indispensable merit of factual knowledge in the inference stage. 3) The performance of KPT exceeds meta-learning (MetaICL) and self-supervised learning (SelfSup.) approaches by around 4%, which are also focused on continual pre-training. This demonstrates that explicitly injecting knowledge into PLM is more effective for ICL, which is imperative and makes a dominant role in ICL. 4) Our method attains more impressive performance when combining all of these knowledgeable techniques, highlighting the necessity of factual knowledge in ICL. We provide a detailed analysis in Section 4.3. 5) We also evaluate the other scales for GPT-2 and OPT in 8-shot settings. Results in Appendix F show that the improvements are consistent in different PLMs.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	27.93±4.8	54.49±3.5	46.93±3.0	40.31±2.7	42.42
CBU (Zhao et al., 2021)	29.88±3.9	55.40±1.8	49.32±4.0	44.05±4.0	44.66
KATE (Liu et al., 2022)	29.02±4.0	55.10±3.9	47.25±3.4	42.77±3.8	43.54
MetaICL† (Min et al., 2022a)	31.16±3.2	55.64±2.9	50.46±2.6	46.72±2.7	46.00
SelfSup.† (Chen et al., 2022a)	31.32±3.0	54.88±3.0	49.97±2.7	47.50±3.5	45.92
KICT†	**36.17±1.8**	**58.11±2.4**	**54.23±2.6**	**50.46±3.3**	**49.74**
only w. KPT†	34.21±4.3	57.32±2.2	52.79±3.0	49.93±1.9	48.56
only w. KER	29.56±2.3	55.82±1.2	48.11±2.4	43.58±2.1	44.27
only w. KCP	33.60±3.7	57.77±2.4	51.63±2.9	46.09±3.1	47.27

Table 2: The 8-shot performance (%) on GPT-2 (large) of different learning settings with standard deviations over question answering benchmarks.

4.3 Ablation Study

We further investigate how these proposed knowledgeable techniques contribute to the final performance with different combinations. As shown in Table 3, results demonstrate that no matter what combination, it greatly promotes the overall performance of vanilla ICL. We also have an interesting observation. KPT is more important for performance improvement, and achieves higher scores than KER and KCP, indicating that the best way of unleashing the PLM power is to inject knowledge into the model parameters. Nonetheless, the combination of KER+KCP also respectively improves ICL by about 8% for each task. This indicates that KER and KCP are also critical to ICL because ultra-large PLM can not be continuously pre-trained or tuned in real-world scenarios to save compuational resources. In addition, results from Table 1 to Ta-

Baselines	SST-2 acc	MRPC f1	MNLI acc	RTE acc	AGNews acc	TREC acc	ComQA acc	Quartz acc	SQuAD em	Quoref em
ICL	76.18±7.2	54.46±2.3	56.85±2.4	53.94±5.0	45.67±6.6	51.56±4.1	27.93±4.8	54.49±3.5	46.93±3.0	40.31±2.7
KPT+KER	91.04±3.3	67.93±3.0	68.47±2.9	61.30±3.3	62.18±3.9	61.52±3.1	35.17±4.0	57.64±2.6	52.23±3.4	50.20±3.1
KPT+KCP	90.65±3.7	68.44±2.5	68.89±3.4	62.38±2.3	63.88±3.5	62.12±2.9	**36.38±2.2**	58.03±2.0	54.17±1.8	50.18±2.2
KER+KCP	86.45±3.0	64.07±2.4	66.60±2.9	57.39±3.2	58.95±3.6	58.60±3.5	34.26±2.2	57.88±3.1	52.20±2.3	47.92±2.7
All (KICT)	**91.21±2.9**	**69.96±0.7**	**69.59±1.0**	**63.74±4.2**	**68.89±5.7**	**63.52±5.5**	36.17±1.8	**58.11±2.4**	**54.23±2.6**	**50.46±3.1**

Table 3: The 8-shot performance (%) of different combinations of the knowledgeable modules.

Methods	SST-2 acc	AGNews acc	TREC acc	ComQA acc	SQuAD em
None (ICL)	76.18±7.2	45.67±6.6	51.56±4.1	27.93±4.8	46.93±3.0
GPT-2	81.35±3.0	48.72±2.7	52.36±3.3	28.61±3.8	47.14±3.1
KPT	**90.04±3.5**	**59.74±4.4**	**60.16±2.0**	**34.21±4.3**	**52.79±3.0**
w/o. MEP	84.40±4.0	51.29±3.9	54.72±3.1	33.01±7.7	52.23±2.8
w/o. EDG	87.19±2.9	56.40±4.3	55.91±3.1	31.95±5.9	50.80±3.9
w/o. KQA	85.30±3.3	53.03±3.6	53.46±2.4	30.08±5.8	49.71±4.6

Table 4: The 8-shot performance (%) of each self-supervised task. GPT-2 denotes the vanilla objective.

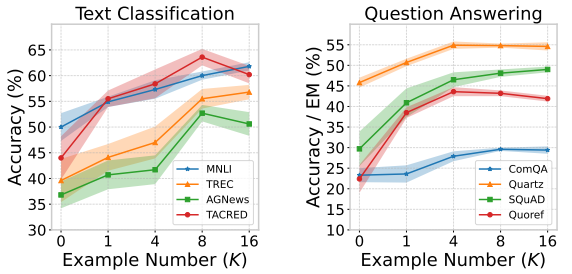


Figure 5: GPT-2 (large) sample effectiveness (%) of KICT (only w. KER) with different values of K.

ble 3 show that our method has improved significantly on classification tasks. We think that the benefits of injecting knowledge over simple language understanding tasks are more obvious than question answering.

4.4 Further Analysis

Effectiveness of KPT. To investigate what makes a high performance for KPT, we test the effectiveness of each knowledgeable self-supervised task. For a fair comparison, we also choose two baselines: 1) **None** is that we do not use any self-supervised task, which is the same as vanilla ICL proposed in (Brown et al., 2020), 2) **GPT-2** represents conventional autoregressive language modeling (ALM) pre-training tasks. As shown in Table 4, KPT can make substantial improvements for ICL. Particularly, all the self-supervised learning tasks in KPT are complementary for pre-training and outperform the baseline with or without the conventional objective of GPT-2. In addition, the

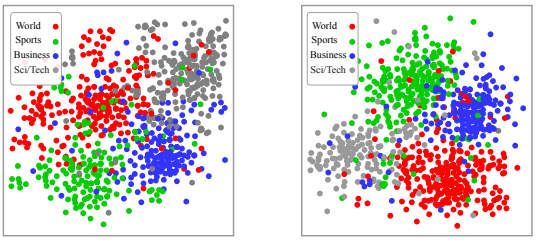


Figure 6: Visualizations of each AGNews's training example. KATE (left) uses CLS embeddings of RoBERTa. Ours (right) utilizes averaged knowledge embeddings.

MEP and KQA tasks are most critical for classification and question answering, respectively, which demonstrates that different pre-training objectives possess different advantages in downstream tasks.

Sample Effectiveness. To investigate the influence of the number of in-context examples K, we choose multiple classification and question answering tasks and vary K from 0, 1, 4, 8 to 16. From Figure 5, we find that increasing K generally helps across both classification and question answering tasks, demonstrating that more in-context examples may bring more knowledge to better guide the PLM to make predictions. When $K > 8$, the performance of the most tasks will decrease, because the maximum length limit causes information loss. The suitable value K is set around 8.

Visualization of Selected Examples in KER. In addition, for explicitly seeing the performance in semantic space, we obtain the t-SNE (Van der Maaten and Hinton, 2008) visualization of each training example over AGNews via averaged representations of all corresponding entities. We choose KATE as our strong baseline, which is also focused on the example selection [9]. Figure 6 demonstrates that our method can build better semantic representations toward factual knowledge.

Permutations of In-Context Examples. We also compare different permutations of these selected

[9]We do not fine-tune RoBERTa on the training set.

Baselines	SST-2	MRPC	MNLI
Random	79.42±2.7	**59.26±2.5**	**59.93±1.0**
Ascending	78.29±2.2	58.05±2.6	59.31±1.5
Descending	**79.61±3.0**	58.16±3.0	59.58±1.3

Table 5: The 8-shot averaged results (%) of KICT (only w. KER) for different permutations.

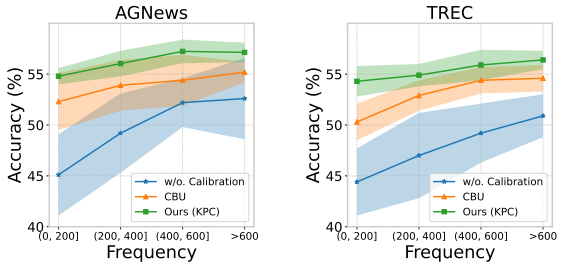


Figure 7: GPT-2 (large) 4-shot performance of calibration over difference word frequencies.

examples according to the sample weight computed in Eq. 3. In Table 5, Random means to randomly choose an order. Ascending and Descending respectively denote that the example order is ascending or descending by the weight. From the results, we find no tangible relationship between the sampling weight and order.

Effectiveness of KPC. We finally conduct analysis on prediction calibration. We choose AGNews and TREC tasks and follow the same settings in the preliminary experiments (we randomly choose two label words from different frequency regions). Results in Figure 7 demonstrate that calibrating the prediction consistently achieve improvements than the vanilla approach. In addition, we find that the prediction results highly depend on the label frequency, which is similar to Figure 3. However, our KPC still outperforms the strong baseline Calibrate Before Use (CBU) with arbitrary label frequency, which only transforms the input into content-free prompts. It underscores that the prior information of each label word in KB is non-negligible. In other words, calibration by the prior information can alleviate the impact of label frequency.

5 Related Work

Pre-trained Language Models (PLMs). Large-scale PLMs aim at learning semantic representations over unsupervised corpora and have made tremendous progress in the NLP community. Notable PLMs can be divided into three main types, including encoder-only (Devlin et al., 2019; Liu et al., 2019; He et al., 2021; Yang et al., 2019; Lan et al., 2020), decoder-only (Radford et al., 2018; Brown et al., 2020; Zhang et al., 2022a) and encoder-decoder (Lewis et al., 2020; Raffel et al., 2020). To inject factual knowledge into PLMs, a branch of knowledge-enhanced PLMs (Zhang et al., 2019; Sun et al., 2020a; Wang et al., 2021b,a, 2022; Pan et al., 2022) have been proposed for PLMs to capture rich semantic knowledge from KBs. Our work focuses on decoder-only models (e.g., GPT-2) and injects factual knowledge to further improve the ICL's performance.

Prompting for PLMs. Prompt-based learning aims to add natural language prompts to guide the PLM to solve downstream tasks. A series of works focus on tunable discrete prompt-tuning (Gao et al., 2021; Raffel et al., 2020) and continuous prompt-tuning (Liu et al., 2021b; Gu et al., 2021). In contrast, GPT-3 (Brown et al., 2020) enables in-context learning (ICL) with a text-based prompt in zero-shot scenarios that bypasses the parameter update (Dong et al., 2023). To explore what facets affect ICL, previous works focus on input-output mapping (Min et al., 2022b; Kim et al., 2022), meta-learning (Chen et al., 2022b; Min et al., 2022a), prompt engineering (Liu et al., 2022, 2021a), prediction calibrating (Zhao et al., 2021; Hu et al., 2022), etc. Recently, Chain-of-Thought (CoT) is presented to leverage reasoning and interpretable information to guide the large PLM to generate reliable responses (Si et al., 2022; Zhang et al., 2022b; Wei et al., 2022). Different from them, we fully exploit *factual knowledge* to better improve ICL in pre-training, prompting, and prediction.

6 Conclusion

In this paper, we explore and exploit *factual knowledge* in ICL, such as inherent knowledge learned in PLM, relevant knowledge derived from the selected training examples, and knowledge biases in prediction. We propose a novel knowledgeable in-context tuning (KICT) framework to further improve the ICL's performance by fully exploiting factual knowledge in the procedures of pre-training, prompting, and prediction. Extensive experiments illustrate that each technique substantially achieves improvements and outperforms the strong baselines over classification and question answering tasks. In the future, we will 1) explore the reasoning and interpretability of knowledge in ICL, and 2) extend our works to encoder-decoder PLMs.

Limitations

We provide some limitations of our work. 1) We only focus on decoder-only PLM because the original in-context learning is mainly focused on decoder-only generation, such as GPT-2, GPT-3, OPT, etc. However, we think it can be extended to the encoder-decoder model, which aims to use for translation, and conditional generation. 2) Due to the computation resources limitation, we ignore the experiment settings for urge-large PLMs over 10B parameters. 3) We focus on investigating factual knowledge in three procedures, i.e., pre-training, prompting, and prediction. However, we believe that the knowledge may have other impact facets, such as reasoning, interpretability, etc. We will leave it as our future work.

Ethical Considerations

Our contribution in this work is fully methodological, namely a knowledgeable in-context tuning (KICT) to boost the performance of PLMs with factual knowledge. However, transformer-based models may have some negative impacts, such as gender and social bias. Our work would unavoidably suffer from these issues. We suggest that users should carefully address potential risks when the KICT models are deployed online.

References

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022a. Improving in-context few-shot learning via self-supervised training. In *NAACL*, pages 3558–3573.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022b. Meta-learning via language model in-context tuning. In *ACL*, pages 719–730.

Pradeep Dasigi, Nelson F. Liu, Ana Marasovic, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *EMNLP*, pages 5924–5931.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, pages 1625–1628.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL*, pages 3816–3830.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. PPT: pre-trained prompt tuning for few-shot learning. *CoRR*, abs/2109.04332.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *ICLR*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *ACL*, pages 2225–2240.

Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. *CoRR*, abs/2205.12685.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *ACL*, pages 100–114.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: enabling language representation with knowledge graph. In *AAAI*, pages 2901–2908.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *ACL*, pages 8086–8098.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. Metaicl: Learning to learn in context. In *NAACL*, pages 2791–2809.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? *CoRR*, abs/2202.12837.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*, pages 3470–3487.

Xiaoman Pan, Wenlin Yao, Hongming Zhang, Dian Yu, Dong Yu, and Jianshu Chen. 2022. Knowledge-in-context: Towards knowledgeable semi-parametric language models. *CoRR*, abs/2210.16433.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*, pages 784–789.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *NAACL*, pages 2655–2671. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan L. Boyd-Graber, and Lijuan Wang. 2022. Prompting GPT-3 to be reliable. *CoRR*, abs/2210.09150.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020a. Colake: Contextualized language and knowledge embedding. In *COLING*, pages 3660–3670.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020b. ERNIE 2.0: A continual pre-training framework for language understanding. In *AAAI*, pages 8968–8975.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. Quartz: An open-domain dataset of qualitative relationship questions. In *EMNLP*, pages 5940–5945.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*, pages 4149–4158.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR*, pages 200–207. ACM.

Jianing Wang, Wenkang Huang, Qiuhui Shi, Hongbin Wang, Minghui Qiu, Xiang Li, and Ming Gao. 2022. Knowledge prompting in pre-trained language model for natural language understanding. *CoRR*, abs/2210.08536.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *ACL*, pages 1405–1418.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *TACL*, 9:176–194.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Jian Yang, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, and Jinghui Peng. 2021. A survey of knowledge enhanced pre-trained models. *CoRR*, abs/2110.00269.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *ACL*, pages 1441–1451.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *CoRR*, abs/2210.03493.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Details of Preliminary Experiments

A.1 Details of Destruction Settings

We choose 8 classification tasks and 4 question answering tasks for the preliminary experiments. The details of these datasets are shown in Appendix B.

To investigate the impact of factual knowledge, we assume that the entities (sometimes with labels) in the example can represent the factual knowledge (Wang et al., 2021b, 2022, 2021a; Sun et al., 2019; Zhang et al., 2019). Thus, we recognize all entities by the open-source entity linking tool TagMe[10] (Ferragina and Scaiella, 2010). For the classification tasks, we also view the labels as special entities.

We follow (Min et al., 2022b; Kim et al., 2022) to design multiple destruction settings which aim to remove or replace all the entities (and the labels) to show the impact of the factual knowledge. Likewise, to make a fair comparison, we also randomly choose some non-entity tokens (the number of these tokens is the same as the number of entity tokens). For each task, we randomly choose $K = 8$ examples as the in-context examples and concatenate them with each test example to form an input sequence. The maximum sequence length of each example is 256. We choose 5 different random seeds (i.e., 12, 24, 42, 90, and 100). Hence, each dataset has 5 different testing results for one PLM. In other words, for each PLM, we can obtain $8 \times 5 = 40$ results for classification and $4 \times 5 = 20$ results for question answering. We thus report the averaged results of each PLM in Figure 2. Through the results, we find that factual knowledge is key to the performance of ICL and is more important than non-entity components.

[10] https://sobigdata.d4science.org/group/tagme.

A.2 Details of Frequency Settings

In the preliminary experiment, we investigate the impact of label word frequency. Specifically, we choose two classic tasks AGNews and TREC. We first randomly choose $K = 4$ examples from the training set to form the in-context prompt. Then, we use the rest of the training examples as target examples to make predictions. We do not use the development or testing sets because the scale of them is too small to demonstrate the frequency obviously. During the prediction, we obtain 4 generated words which have top-4 highest prediction probabilities. Thus, we can calculate the frequency statistics of each generated word. Due to the space limitation, we only show the top-8 label words statistics of each category over AGNews in Figure 8.

To investigate the impact of the frequency, for each frequency region (i.e., $(0, 200]$, $(200, 400]$, $(400, 600]$, > 600), we randomly choose two label words for the prediction. For example, we can choose "teams" and "groups" from the frequency region > 600 for the label "sports" in the AGNews task. Thus, we can respectively obtain $2^4 = 16$ and $2^6 = 64$ permutations for AGNews and TREC. We choose GPT-2 (urge) with 1.5B parameters and report the average results and show the box plots in Figure 3.

A.3 Analysis of the Knowledge Relevance in In-Context Examples

In the preliminary experiments, we find that the factual knowledge in the selected in-context examples

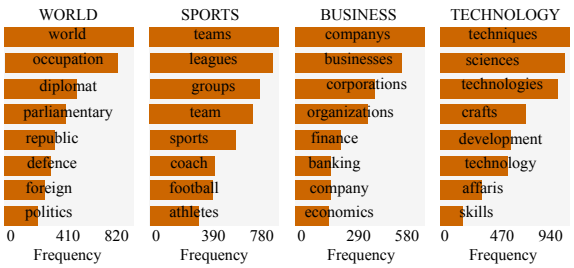


Figure 8: The label words frequency of AGNews.

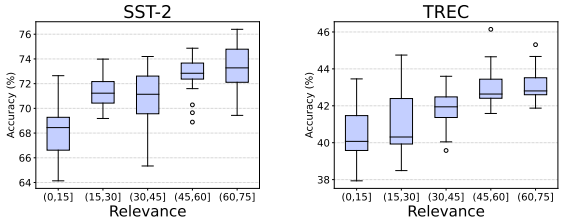


Figure 9: The 4-shot performance (%) with different knowledge relevance over SST-2 and TREC.

is key to ICL. To further validate these findings, we choose two datasets for further analysis, including SST-2 and TREC.

Specifically, we use the proposed technique in knowledgeable example retrieval (KER) to obtain the knowledge relevance score for each training example. Thus, for each score region (i.e., $(0, 15]$, $(15, 30]$, $(30, 45]$, $(45, 60]$, $(60, 75]$), we can sample $K = 4$ examples for the in-context prompt. For each region, we can obtain the average performance on $4 \times 3 \times 2 \times 1 = 24$ orders. We draw the box plot in Figure 9. Results show that the selected examples with higher knowledge relevance make consistent contributions to ICL, indicating that the factual knowledge in the selected examples is critical to ICL.

B Details of the Corpus and Downstream Benchmarks

B.1 Corpora and Knowledge Base

We propose knowledgeable pre-training (KPT), which is similar to the current flourishing research of *knowledge-enhanced pre-trained language models* (KEPLMs) (Liu et al., 2020; Sun et al., 2019, 2020b; Wang et al., 2022). Different from them, we focus on auto-regressive PLMs, such as GPT-2. We collect training corpora from Wikipedia (2020/03/01)[11], and use WikiExtractor[12] to process the pre-training data. The knowledge base (KB) $\mathcal{G}$ we choose is WikiData5M (Wang et al., 2021b), which is an urge-large structural data source based on Wikipedia. The entity linking toolkit we used is TagMe. In total, we have 3,085,345 entities and 822 relation types in $\mathcal{G}$, and 25,933,196 training sentences.

As mentioned above, KPT consists of three self-training tasks, i.e., *masked entity prediction*, *entity description generation*, and *knowledge question answering*. For each task, we randomly select multiple sentences to form a training instance until reaching the maximum sequence length (i.e., 2048). Finally, we have sampled 100k training instances for each task. In average, we have 8 examples for each instance.

B.2 Downstream Task Datasets

To evaluate the effectiveness of our framework, we choose 8 text classification tasks and 4 question answering tasks. For the text classification, we directly choose 8 tasks from (Gao et al., 2021; Zhao et al., 2021). All the classification tasks involve sentiment analysis, natural language inference (NLI), question classification, and topic classification. For the question answering tasks, we choose four widely used tasks, including CommonsenseQa (ComQA) (Talmor et al., 2019), Quartz (Tafjord et al., 2019), SQuAD (Rajpurkar et al., 2018) and Quoref (Dasigi et al., 2019), where ComQA and Quartz are multi-choice QA, SQuAD and Quoref are extractive QA. The statistics of each dataset are shown in Table 6.

C Implementation Details

C.1 Pre-training Details

In the pre-training stage, we choose different scales of GPT-2 (0.1B, 0.3B, 0.8B, 1.5B) (Brown et al., 2020) and OPT (Zhang et al., 2022a) (2.7B, 6.7B) from HuggingFace[13] as the underlying PLMs. We do not use larger GPT-3 models because of the computation resource limitations. Because all three kinds of pre-training tasks share the same format, we can directly mix up all the pre-training examples to form a cross-task pre-training paradigm. We

[11] https://dumps.wikimedia.org/enwiki/.

[12] https://github.com/attardi/wikiextractor.

[13] https://huggingface.co/transformers/index.html.

Category	Dataset	#Class	#Train	#Test	Type	Labels (classification tasks)
Text Classification	SST-2	2	6,920	872	sentiment	positive, negative
	MRPC	2	3,668	408	paraphrase	equivalent, not_equivalent
	MNLI	3	392,702	9,815	NLI	entailment, neutral, contradiction
	QNLI	2	104,743	5,463	NLI	entailment, not_entailment
	RTE	2	2,490	277	NLI	entailment, not_entailment
	CB	3	250	57	NLI	entailment, neutral, contradiction
	TREC	6	5,452	500	question cls.	abbr., entity, description, human, loc., num.
	AGNews	4	120,000	7,600	topic cls.	world, sports, business, technology
Question Answering	ComQA	-	9,741	1,221	multi-choice	-
	Quartz	-	2,696	384	multi-choice	-
	SQuAD	-	87,599	10,570	extractive QA	-
	Quoref	-	19,399	2,418	extractive QA	-

Table 6: The statistics of multiple text classification and question answering datasets. Since the original test data is unavailable, we use the development sets as our test sets.

find that it is suitable for the PLM to learn cross-task knowledge. We train our model by AdamW algorithm with $\beta_1 = 0.9, \beta_2 = 0.98$. The learning rate is set as 1e-5 with a warm-up rate 0.1. We also leverage dropout and regularization strategies to avoid over-fitting. The models are trained on 8 NVIDIA A100-80G GPUs.

C.2 Prompting Details

We describe the implementation details with knowledgeable example retrieval (KER). Given a training dataset and a testing set, we aim to choose K examples from the training set which have a high knowledge relevant to all testing examples. To reach this goal, we utilize both Jaccard similarity and Euclidean distance in terms of pre-trained knowledge embeddings. For pre-trained knowledge embeddings, we choose the ConVE (Dettmers et al., 2018) algorithm to pre-train over wikidata5m and obtain the embeddings of entities and relations. We set its dimension as 768, the negative sampling size as 64, the batch size as 128 and the learning rate as 0.001. Finally, we only store the embeddings of all the entities. The KER algorithm for the prompting is shown in Algorithm 1.

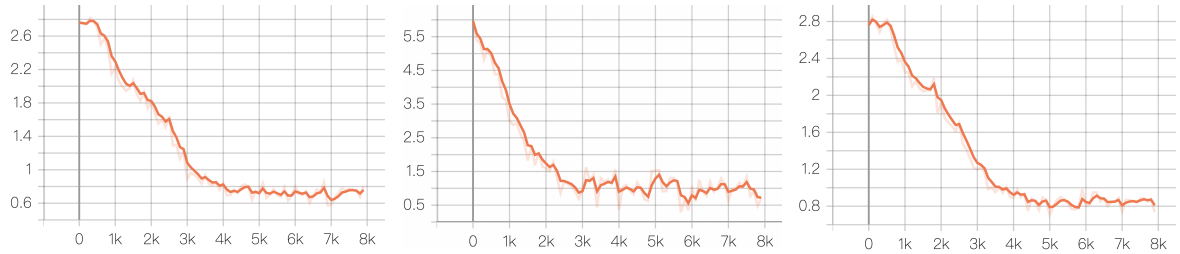
C.3 Prediction Details

We first provide the details of the prompt formats and label mapping rules. Specifically, for the classification task, we need to define a template and label mapping to guide the model to generate results toward pre-defined classes. The prompt formats and label words are shown in Table 8. For the question answering task, we only need to define the template format, shown in Table 9.

Algorithm 1 Knowledgeable Example Retrieval

Require: Training set $\mathcal{D}_{trn}$, Target (testing) set $\mathcal{D}_{tgt}$, number of in-context examples K.
1: Randomly sampling a subset $\mathcal{D}'_{trn}$ from $\mathcal{D}_{trn}$;
2: **for** each target example $(X_j^{tgt}) \in \mathcal{D}_{tgt}$ **do**
3: Extract entities E_j^{tgt} from this target example;
4: **for** each training example $(X_i^{trn}, y_i^{trn}) \in \mathcal{D}'_{trn}$ **do**
5: Extract entities E_i^{trn} from this training example;
6: Calculate Jaccard similarity $d_{jac}(i,j)$ and Euclidean distance $d_{sem}(i,j)$;
7: **end for**
8: Conditioning on the target example X_j^{tgt}, obtain the knowledge relevance score $d(X_i^{trn}, X_j^{tgt})$ for the training example X_i^{trn} in Eq. 2;
9: **end for**
10: Calculate the final sampling weight $s'(X_i^{trn})$ for each training example X_i^{trn} in Eq. 3;
11: Sampling K training examples via the weight $s'(X_i^{trn})$;
12: **return** The selected K training examples.

During the prediction, we calibrate the prediction probability. We thus provide the implementation details. We obtain a subset of training corpora from the KQA pre-training task, which consists of many question answer pairs. Thus, for each question, we can generate an answer (may be an entity or a label word) at the output position, and obtain the contextualized prior via Eq. 6. The value $P(v)$ means the prior information of the generated entity or label word. Intuitively, if the value $P(v)$ is higher, the entity or label word v is more likely to be generated. We can save these prior values before prediction for downstream tasks. During the prediction, we can use the prior information of each pre-defined label word or entity to calibrate the prediction probability via Eq. 7.



(a) Masked Entity Prediction (b) Entity Description Generation (c) Knowledge Question Answering

Figure 10: The curves of the pre-training loss on GPT-2 (large) for each self-supervised learning task.

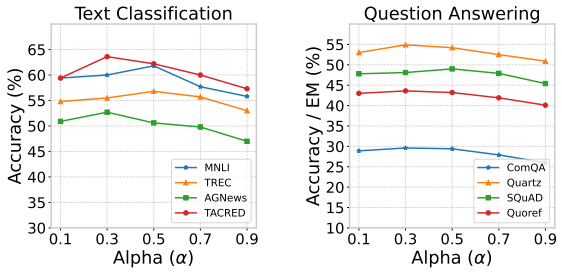


Figure 11: The 8-shot performance (%) of GPT-2 (large) with different α over text classification and question answering tasks.

Hyper-parameter	Value
Batch Size	{2, 4, 8, 16, 32, 64}
Seed	{12, 24, 42, 90, 100}
K	{0, 1, 4, 8, 16}
α	{0.1, 0.3, 0.5, 0.7, 0.9}
γ	{0.001, 0.01, 0.05, 0.1, 0.5, 1.0}

Table 7: The searching scope for each hyper-parameter.

D Analysis of Settings of Model Variants

We conduct some detailed analysis of our proposed technique.

Analysis of Pre-training Efficiency. To show the efficiency of pre-training, we choose GPT-2 (large) draw the pre-training loss for each self-supervised learning task. From Figure 10, we can see that as the training process proceeds, each self-supervised learning task has reached the convergence of the model through the entire pre-training process.

Effectiveness of Hyper-parameters. In KICT, we investigate the effectiveness of the hyper-parameter α in KER, which aims to balance the relevance scores between Jaccard similarity and Euclidean distance. Results shown in Figure 11 demonstrate that the hyper-parameter α is key to the performance. We can see that the suitable value is around 0.3.

Effectiveness of the Template. We believe that the model performances rely on the format of the template, which has been investigated in (Liu et al., 2022; Min et al., 2022b). We choose some other templates for evaluation. For example, when we change the prefix string (e.g., "Question:", "Answer:") to others (e.g., "Q:", "A:"), the performance improvement of KICT is consistent. In addition, we also find that the text split character "\n" between each sentence or example is important to support the generation, which is also found in (Dong et al., 2023; Andrew and Gao, 2007; Kim et al., 2022; Si et al., 2022).

E Details of the Grid Search

For the downstream task inference, the searching scope of each model hyper-parameter is shown in Table 7.

F Performance on Different PLMs

To show that our method is general and can be applied to other similar models, we choose other scale sizes of GPT-2 and OPT to show the effectiveness of our KICT. More other experiments results are shown from Table 10 to Table 17.

Task	Prompt	Label Words
SST-2	Review: This movie is amazing! Sentiment: Positive Review: Horrific movie, don't see it. Sentiment:	Positive, Negative
MRPC	Whether the two questions are similar? Question 1: How much is this book? Question 2: How many books? Output: No Question 1: Do you know the reason? Question 2: What's the reason? Output:	Yes, No
MNLI	Is entailment, neutral, or contradiction between two texts? Text 1: We sought to identify practices within the past 5 years. Text 2: We want to identify practices commonly used by agencies in the last 5 years. Output: entailment Text 1: yeah well you're a student right Text 2: Well you're a mechanics student right? Output:	entailment, neutral, contradiction
QNLI	Whether the answer is entailed to the question? Text 1: In what year did the university first see a drop in applications? Text2: In the early 1950s, student applications declined as a result of increasing crime and $\cdots$ Output: Yes Text1: When did Tesla move to Gospic? Text2: Tesla was the fourth of five children. Output:	Yes, No
RTE	Others argue that Mr. Sharon should have negotiated the Gaza pullout - both to obtain at least some written promises of $\cdots$ Question: Mr. Abbas is a member of the Palestinian family. True or False? Answer: False The program will include Falla's "Night in the Gardens of Spain," Ravel's Piano $\cdots$ Question: Beatrice and Benedict is an overture by Berlioz. True or False? Answer:	True, False
CB	But he ended up eating it himself. I was reluctant to kiss my mother, afraid that somehow her weakness and unhappiness would infect me. $\cdots$ Question: her life and spirit could stimulate her mother. True, False, or Neither? Answer: Neither Valence the void-brain, Valence the virtuous valet. Why couldn't the figger choose his own portion of titanic anatomy to shaft? Did he think he was helping? Question: Valence was helping. True, False, or Neither? Answer:	True, False, Neither
TREC	Classify the questions based on whether their answer type is a Number, Location, Person, Description, Entity, or Abbreviation. Question: How did serfdom develop in and then leave Russia? Answer Type: Description Question: When was Ozzy Osbourne born? Answer Type:	Number, Location, Person, Description, Entity, Abbreviation
AGNews	Article: USATODAY.com - Retail sales bounced back a bit in July, and new claims for jobless benefits fell last week, the government said Thursday, indicating $\cdots$ Answer: Business Article: New hard-drive based devices feature color screens, support for WMP 10. Answer:	World, Sports, Business, Technology

Table 8: The prompts used for text classification. We show one training example per task for illustration purposes. The right column shows the label words (aiming to map the word to the original label class).

Task	Prompt
ComQA	Answer the question through multiple-choice. Question: When people want to watch a new move, the often go see it at the? (A) town (B) conference (C) bathroom (D) theater (E) train station Answer: theater Question: Where is known to always have snow? (A) africa (B) north pole (C) roof (D) canada (E) surface of earth north pole Answer:
Quartz	Answer the question through multiple-choice. Question: Eric pushes an electron closer to the nucleus of an atom. The electron _____ energy.As you go farther from the nucleus of an atom, the electron levels have more and more energy. (A) loses (B) gains Answer: gains Question: When something is very lightweight what does it need to move?Objects with greater mass have greater inertia. (A) more inertia (B) less inertia Answer:
SQuAD	Read the question and find an answer in the context. Question: Where was the first figure skating championship held? Context: The tourism industry began in the early 19th century when foreigners visited the Alps, traveled to the bases of the mountains to enjoy the scenery, and stayed at the spa-resorts. Large hotels were built during the Belle Époque; cog-railways, built early in the 20th century, brought tourists to ever higher elevations, with the Jungfraubahn terminating at the Jungfraujoch, well above the eternal snow-line, after going through a tunnel in Eiger. During this period winter sports were slowly introduced: in 1882 the first figure skating championship was held in St. Moritz, and downhill skiing became a popular sport with English visitors early in the 20th century, as the first ski-lift was installed in 1908 above Grindelwald. Answer: St. Moritz Question: What are some examples of classical violinists from Portugal? Context: In the classical music domain, Portugal is represented by names as the pianists Artur Pizarro, Maria João Pires, Sequeira Costa, the violinists Carlos Damas, Gerardo Ribeiro and in the past by the great cellist Guilhermina Suggia. Notable composers include José Vianna da Motta, Carlos Seixas, João Domingos Bomtempo, João de Sousa Carvalho, Luís de Freitas Branco and his student Joly Braga Santos, Fernando Lopes-Graça, Emmanuel Nunes and Sérgio Azevedo. Similarly, contemporary composers such as Nuno Malo and Miguel d'Oliveira have achieved some international success writing original music for film and television. Answer:
Quoref	Read the question and find an answer in the context. Question: What's the name of the person whose birth causes Sarah to die? Context: Jack and Sarah are expecting a baby together, but a complication during the birth leads to the death of Sarah. Jack, grief-stricken, goes on an alcoholic bender, leaving his daughter to be taken care of by his parents and Sarah's mother, until they decide to take drastic action: they return the baby to Jack whilst he is asleep, leaving him to take care of it. · · · Answer: Sarah Question: What is the first name of the person the actor believes is a little too odd? Context: When a British secret agent is murdered in the line of duty, agent Karen Bentley inherits the mission from her partner. The mission is to deliver a flight plan for a hundred American bomber planes to a British agent in Chicago. The plans are hidden in a small medallion of a scorpion that Karen wears. · · · Answer:

Table 9: The prompts used for question answering. We show one training example per task for illustration purposes.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	66.58±4.7	44.73±2.5	49.80±2.9	46.33±2.2	45.70±3.8	36.92±2.3	44.38±2.6	40.53±4.0	46.87
CBU (Zhao et al., 2021)	74.19±4.1	48.88±3.3	51.10±2.5	48.39±3.2	40.07±3.0	39.26±2.8	47.94±2.2	43.28±2.2	49.14
KATE (Liu et al., 2022)	72.38±2.9	46.38±3.2	49.15±3.0	47.28±2.8	46.30±2.6	41.48±2.1	47.80±2.2	43.83±3.1	49.95
MetaICL† (Min et al., 2022a)	77.20±3.6	51.21±2.5	53.29±3.0	49.42±2.2	48.33±2.0	40.18±1.9	49.68±2.8	47.35±2.9	52.08
SelfSup.† (Chen et al., 2022a)	78.94±3.0	52.13±2.0	52.70±2.2	48.29±1.8	49.27±2.6	41.80±2.5	48.59±2.5	47.39±3.2	52.39
KICT†	**82.18±3.2**	**54.19±3.7**	**54.85±2.3**	**50.93±1.9**	**50.13±2.2**	**43.89±2.8**	**51.38±2.5**	**51.20±3.0**	**54.90**

Table 10: The 8-shot performance (%) on GPT-2 (small) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	71.39±3.2	49.60±2.8	53.90±2.4	50.04±3.2	51.18±4.1	39.33±2.8	49.20±2.1	43.75±3.6	51.05
CBU (Zhao et al., 2021)	77.71±3.8	55.48±3.1	55.41±2.2	51.10±3.0	47.53±2.8	48.11±2.7	51.52±2.7	53.27±2.4	55.02
KATE (Liu et al., 2022)	75.32±3.1	53.80±3.1	48.88±3.4	50.14±2.5	45.82±2.9	47.05±2.4	50.25±2.8	51.93±3.4	52.89
MetaICL† (Min et al., 2022a)	80.16±3.0	61.33±2.0	56.12±3.1	54.24±2.9	54.93±2.9	46.50±2.9	53.22±2.8	53.36±2.4	57.48
SelfSup.† (Chen et al., 2022a)	81.62±3.0	58.43±3.2	59.53±2.6	51.70±3.8	54.33±2.6	43.48±3.5	53.46±2.6	53.73±3.1	57.04
KICT†	**89.10±3.9**	**66.44±2.7**	**64.85±3.0**	**57.81±3.2**	**61.02±4.0**	**53.91±2.3**	**60.34±2.0**	**61.77±3.3**	**64.41**

Table 11: The 8-shot performance (%) on GPT-2 (medium) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	78.98±7.2	56.36±2.3	58.25±2.4	55.03±3.2	55.01±5.0	44.04±1.8	53.29±4.1	47.33±6.6	56.04
CBU (Zhao et al., 2021)	83.31±4.4	65.17±3.9	58.13±2.8	55.59±3.9	55.97±2.8	53.14±1.7	56.29±3.7	57.89±2.8	60.69
KATE (Liu et al., 2022)	82.55±3.8	59.43±3.9	61.20±2.4	55.37±3.5	55.57±2.7	48.27±2.1	56.11±2.8	53.78±3.4	59.04
MetaICL† (Min et al., 2022a)	88.80±5.0	64.22±2.0	62.39±3.4	57.34±1.9	59.18±2.8	50.46±2.5	57.90±1.8	57.13±2.4	62.18
SelfSup.† (Chen et al., 2022a)	88.55±3.0	64.24±2.0	63.42±2.2	55.70±1.8	58.93±2.6	48.08±2.5	58.01±2.5	58.28±3.2	61.90
KICT†	**92.18±2.9**	**71.32±0.7**	**71.23±1.0**	**62.89±1.2**	**66.10±4.2**	**58.33±3.8**	**64.90±5.5**	**69.27±5.7**	**69.53**

Table 12: The 8-shot performance (%) on GPT-2 (urge) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	79.43±7.2	56.72±2.3	59.28±2.4	55.37±3.2	56.01±5.0	44.48±1.8	54.10±4.1	47.95±6.6	56.67
CBU (Zhao et al., 2021)	83.77±4.4	65.38±3.9	58.49±2.8	55.88±3.9	56.26±2.8	53.89±1.7	56.37±3.7	58.20±2.8	61.03
KATE (Liu et al., 2022)	83.18±3.8	59.83±3.9	62.40±2.4	55.87±3.5	55.81±2.7	48.83±2.1	56.98±2.8	54.32±3.4	59.65
MetaICL† (Min et al., 2022a)	90.03±5.0	64.72±2.0	62.99±3.4	57.94±1.9	59.81±2.8	51.29±2.5	58.50±1.8	58.12±2.4	62.93
SelfSup.† (Chen et al., 2022a)	88.59±3.0	64.24±2.0	64.42±2.2	56.60±1.8	59.22±2.6	49.58±2.5	59.33±2.5	59.48±3.2	62.77
KICT†	**92.38±2.9**	**71.92±0.7**	**71.83±1.0**	**63.21±1.2**	**66.83±4.2**	**58.70±3.8**	**65.38±5.5**	**70.42±5.7**	**70.08**

Table 13: The 8-shot performance (%) on OPT (large) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	23.70±3.7	49.20±1.9	43.10±3.4	37.30±3.0	38.34
CBU (Zhao et al., 2021)	26.37±3.1	52.90±2.8	46.88±2.0	41.38±2.9	41.89
KATE (Liu et al., 2022)	26.89±3.2	52.88±3.1	46.93±3.7	41.35±2.8	42.01
MetaICL† (Min et al., 2022a)	27.40±2.7	52.74±3.3	46.63±2.9	42.51±3.0	42.32
SelfSup.† (Chen et al., 2022a)	27.33±3.1	52.91±3.1	46.97±2.9	42.71±3.2	42.48
KICT†	**28.78±2.6**	**53.10±2.9**	**47.72±2.3**	**43.88±2.2**	**43.37**

Table 14: The 8-shot performance (%) on GPT-2 (small) of different learning settings with standard deviations over question answering benchmarks.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	25.38±3.1	52.10±3.2	45.58±3.3	38.47±2.7	40.38
CBU (Zhao et al., 2021)	28.40±3.2	53.64±2.6	47.81±4.0	43.20±2.2	42.68
KATE (Liu et al., 2022)	28.38±3.1	54.26±3.3	46.70±3.7	41.98±4.1	42.83
MetaICL† (Min et al., 2022a)	29.67±2.9	54.37±2.5	48.79±2.4	45.11±3.1	44.49
SelfSup.† (Chen et al., 2022a)	29.36±3.0	54.10±2.2	48.47±2.7	44.06±3.1	44.00
KICT†	**34.81±3.0**	**56.38±2.9**	**51.18±2.8**	**46.00±3.5**	**47.09**

Table 15: The 8-shot performance (%) on GPT-2 (medium) of different learning settings with standard deviations over question answering benchmarks.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	29.15±2.4	55.78±3.1	49.12±3.1	42.11±2.7	44.04
CBU (Zhao et al., 2021)	31.58±3.9	57.01±2.6	51.28±2.8	45.70±4.4	46.39
KATE (Liu et al., 2022)	31.18±4.1	56.70±3.0	49.13±3.4	44.54±3.3	45.39
MetaICL† (Min et al., 2022a)	32.16±3.2	57.64±2.6	53.26±3.1	48.91±2.9	47.99
SelfSup.† (Chen et al., 2022a)	33.44±3.2	56.18±3.5	51.90±2.7	49.10±3.1	47.66
KICT†	**37.05±2.8**	**59.35±2.4**	**55.08±2.9**	**53.18±3.2**	**51.17**

Table 16: The 8-shot performance (%) on GPT-2 (urge) of different learning settings with standard deviations over question answering benchmarks.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	30.42±2.2	56.19±3.2	48.73±3.0	44.18±3.7	44.88
CBU (Zhao et al., 2021)	32.16±2.7	58.02±2.8	53.11±2.7	47.35±2.0	47.66
KATE (Liu et al., 2022)	33.32±3.6	58.90±2.9	50.65±2.4	46.12±3.5	47.25
MetaICL† (Min et al., 2022a)	33.96±3.4	58.64±2.4	54.11±2.4	48.12±2.7	48.71
SelfSup.† (Chen et al., 2022a)	34.42±3.0	58.12±3.0	54.92±2.7	49.53±1.8	49.25
KICT†	**39.22±2.8**	**61.71±2.4**	**59.67±2.1**	**54.40±3.1**	**53.75**

Table 17: The 8-shot performance (%) on OPT (large) of different learning settings with standard deviations over question answering benchmarks.